

Programowanie obiektowe

Programowanie strukturalne w Pythonie, cz. 2

Roman Putanowicz

Roman.Putanowicz@L5.pk.edu.pl



Instytut Technologii Informatycznych w Inżynierii Lądowej
Wydział Inżynierii Lądowej Politechniki Krakowskiej

Marzec 2017

Sprawdzanie poprawności kodu

- ▶ PyChecker (pychecker)
- ▶ PyLint (pylint)
- ▶ Pyflakes (pyflakes)

PEP – Python Enhancement Proposal

PEP8 - <https://www.python.org/dev/peps/pep-0008/>

Sprawdzanie kodu on-line: <http://pep8online.com/>

Formatowanie kodu

```
// C/C++  
if (x > y) {  
    x = 1;  
    y = 2;  
}
```

Python

```
if x > y  
    x = 1  
    y = 2
```

```
def my_function():  
    for x in an_iterator:  
        print x  
    print("end of writing")
```

```
def another_function_block():
```

Typy danych: typy liczbowe

```
i = 1234  
print(type(i),i)
```

```
big = 2**200  
print(type(big),big)
```

```
octal = 0o17  
print(type(octal), octal)
```

```
hex = 0xaa  
print(type(hex), hex)
```

```
bin = 0b101  
print(type(bin), bin)
```

```
<class 'int'> 1234
```

```
<class 'int'> 1606938044258
```

```
<class 'int'> 15
```

```
<class 'int'> 170
```

```
<class 'int'> 5
```

Typy danych: liczby rzeczywiste i zespolone

```
f = 4.1e210  
print(type(f), f)
```

```
<class 'float'> 4.1e+210
```

```
z = 3+4j  
print(type(z), z)
```

```
<class 'complex'> (3+4j)
```

Typy danych: liczby rzeczywiste o stałej precyzji

```
import decimal
from decimal import Decimal

d = 0.1 + 0.1 + 0.1 - 0.3
print(type(d), d)

<class 'float'> 5.551115123125783e-17

d = Decimal('0.1') + Decimal('0.1') + Decimal('0.1') -
print(type(d), d)

<class 'decimal.Decimal'> 0.0

decimal.getcontext().prec = 4
print("floating division ", 1/7)
print("decimal division ", Decimal(1) / Decimal(7))

floating division  0.14285714285714285
decimal division  0.1429
```

Typy danych: liczby wymierne

```
from fractions import Fraction
x = Fraction(1,3)
y = Fraction(4,6)
print(type(y), y)
```

```
<class 'fractions.Fraction'> 2/3
```

```
print(Fraction('1.25'))
```

```
5/4
```

```
print(Fraction(2/6))
```

```
6004799503160661/18014398509481984
```

Typy danych: typ logiczny

```
flag = True
print(type(flag), flag)           <class 'bool'> True

flag = False
```


Funkcje matematyczne

```
import math
print(math.pi)
print(math.sqrt(2))
print(math.sin(math.pi/2))
```

```
from math import *
print(pi)
print(sqrt(2))
print(sin(pi/2))
```

Typy danych: napisy

```
pc = u'półciężarówka'  
print(type(pc),pc)  
c = pc[3:]  
print(c)  
print(pc[0])  
print(pc[-1])  
print(pc[0:3])  
print(len(pc))
```

```
<class 'str'> półciężarówka  
ciężarówka  
p  
a  
pół  
13
```

Napisy podstawianie

```
pc = u'półciężarówka'  
PC = pc.replace('ówka', 'owa')  
print(pc)  
print(PC)
```

```
półciężarówka  
półciężarówa
```

Napisy wieloliniowe

```
msg = """To jest bardzo
długi tekst, który
jest podzielony na
kilka linii"""
```

```
print(msg)
print(msg.count('\n'))
```

Napisy: łączenie i rozbijanie

```
names = 'apple', 'jabłko', 'apfel'  
print('=' .join(names))
```

```
apple=jabłko=apfel
```

```
line = '1, 2, 3, 4'  
items = [x.strip() for x in line.split(',')]  
print(items)
```

```
['1', '2', '3', '4']
```

Dziękuję za uwagę



Copyrights



Copyright (CC-BY) 2017 Roman Putanowicz
Roman.Putanowicz@L5.pk.edu.pl

This work is licensed under the Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.